DOI: 10.20998/2079-0023.2026.01.09
UDC 004.41 + 004.738.5 + 519.816

D. K. CHUMACHENKO, PhD Student, Software Engineering Department, Odesa Polytechnic National University, 1 Shevchenka av., Odesa, 65044, Ukraine; e-mail: chumachdk@stud.op.edu.ua; ORCID: https://orcid.org/0009-0000-1477-534X
V. V. LIUBCHENKO, Doctor of Engineering, Professor in a Software Engineering Department, Odesa Polytechnic National University, 1 Shevchenka av., Odesa, 65044, Ukraine; e-mail: lvv@op.edu.ua; ORCID: https://orcid.org/0000-0002-4611-7832

RESPONSIBILITY-WEIGHTED AGGREGATION OF QUALITY CRITERIA IN MULTI-LAYER IOT SOFTWARE ARCHITECTURES

Additive quality models are widely used in architectural evaluation because they are transparent, computationally simple, and suitable for integration into ranking and optimization procedures. However, in multi-layer Internet-of-Things systems, directly summing the Edge, Fog, and Cloud contributions introduces a structural bias: the resulting score depends not only on criterion fulfillment but also on the number of layers in which the criterion is realized. This leads to inter-layer double-counting, destroys the unified interpretation scale of criteria, and complicates cross-criterion and cross-scenario comparison. To address this problem, the paper introduces the layer-responsibility matrix ***W***, which distributes the total responsibility for each quality criterion across architectural layers. A corrected aggregation formula is derived as a weighted sum of normalized layer-level contributions under row normalization of ***W***. The paper also provides a lightweight elicitation procedure that allows architects to instantiate ***W*** from scenario characteristics, control logic, dominant risks, and computational placement. The basic properties of the proposed formalism are established, including non-negativity, boundedness, invariance with respect to the number of layers, scenario adaptivity, and interpretability. A numerical example demonstrates how the proposed mechanism eliminates inflated criterion values while preserving the linearity of aggregation. A decision-level numerical example further shows that responsibility-weighted aggregation can reverse the ranking of candidate portfolios and thereby change the architectural decision outcome. The approach is further illustrated through two case studies, infrastructure monitoring and control, and bionic prosthesis software, showing that the same aggregation rule remains valid across domains, whereas responsibility distributions vary according to domain logic. The results justify treating ***W*** as an independent component of formal decision support for IoT software architecture.

Keywords: IoT software architecture; additive quality model; double-counting; layer-responsibility matrix; Edge-Fog-Cloud; multi-layer systems; quality criteria aggregation.

Introduction. Modern Internet-of-Things systems combine peripheral devices, intermediate processing nodes, and cloud infrastructure. In practical deployments, this translates into multiple execution layers: Edge, Fog, and Cloud, each contributing to the fulfillment of non-functional requirements [1]. Architectural evaluation of such systems often relies on additive models, in which the contributions of individual components, patterns, or layers are summed across selected quality criteria. This focus on quality attributes and their trade-offs in Edge-based IoT systems is also reflected in recent mapping studies [2].

Additive models are attractive due to their transparency and computational simplicity. Additivity enables reproducible comparison of alternatives, explanation of results to stakeholders, and integration into multi-criteria optimization procedures. However, in a multi-layer environment without an explicit semantic alignment mechanism, plain summation ceases to be a neutral mathematical operation: it systematically favors architectures that distribute functions across more layers.

The problem is not the existence of multiple layers per se, but that the same quality criterion may be accounted for by contributions from several layers simultaneously. For example, security in an IoT system is supported by local device-level authentication mechanisms, gateway-level filtering, and centralized access-control policies in the Cloud. Summing these contributions without adjustment increases the integral score for “security” as the number of engaged layers increases, even though the underlying requirement remains the same.

In the authors’ previous work, IoT architecture design was formulated as a multi-objective pattern-portfolio synthesis problem with Pareto optimization [3]. In that framework, $P = \{p_1, p_2, \dots, p_n\}$ denotes a finite catalog of candidate architectural patterns, and $K = \{k_1, \dots, k_m\}$ is the set of non-functional quality criteria. A candidate architecture is represented as a binary portfolio vector $\boldsymbol{x}$, where $x_j = 1$ if pattern p_j is selected. The contribution matrix $\boldsymbol{B}(N \times M)$ encodes the normalized score $c_{j,k}$ of pattern p_j toward criterion *k*. The quality of a portfolio is evaluated using an additive objective vector $F(\boldsymbol{x}) = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{x}$, and the solution is a discrete Pareto set of non-dominated portfolios subject to linear feasibility constraints (budget, dependencies, incompatibilities, cardinality). The layer-responsibility matrix introduced in the present paper extends this formalism by distributing each criterion's total weight across architectural layers, eliminating the inter-layer double-counting that arises when Edge, Fog, and Cloud contributions are summed directly. That line of research provided a decision-support mechanism for generating feasible [4], non-dominated architectural portfolios, but it did not isolate the specific semantic problem that arises when additive criterion contributions are distributed across multiple architectural layers.

The contribution of the present paper is not a new Pareto-synthesis procedure as such, but a formal layer-level reconciliation mechanism for additive quality models in multi-layer IoT systems. Specifically, the paper introduces the layer-responsibility matrix ***W***, proves its basic properties, derives the corrected aggregation formula for criterion contributions across Edge, Fog, and Cloud, and demonstrates on a decision-level example that responsibility-weighted aggregation can alter the ranking of architectural portfolios and lead to a more adequate architectural choice.

Related Work and the Double-Counting Problem. In conventional criteria-based models of software architecture, the integral score is typically formed as a sum or weighted sum of element contributions for each criterion. This approach is natural when considering independent alternatives or single-layer configurations, but it proves insufficient for Edge-Fog-Cloud systems. The reason is that in a multi-layer environment, the same requirement is realized in a distributed fashion, so the set of contributions possesses not only a quantitative but also a structural dimension.

Consider a criterion k, for instance, security or low latency. If the evaluation of architecture S is performed separately for each layer and then summed without reconciliation, the maximum aggregated score of k begins to depend not only on the quality of the solution but also on the number of layers in which k is represented. In effect, the criterion loses its fixed scale: for one scenario, it varies within [0, 1], while for another it may range up to $[0, |L|]$, where $|L|$ is the number of layers.

In the broader literature on software architecture evaluation, direct methodological predecessors of the present work include architecture trade-off analysis and multi-criteria decision making. In particular, the CBAM line of work [5] treats architectural decisions through cost–benefit reasoning over quality attributes, while AHP-based approaches [6] formalize the derivation of criterion priorities and rankings of design alternatives from expert judgments. In the IoT domain, trade-off-aware architectural tactics [7] have recently been proposed to analyze security decisions alongside their impact on other quality attributes, rather than in isolation. From the MCDM perspective, studies of weighted-average normalization [8] further show that additive aggregation requires an explicit normalization step in order to transform heterogeneous criterion values into a comparable scale.

However, these approaches address an adjacent rather than an identical problem. They explain how to analyze trade-offs between criteria, how to derive weights, and how to normalize heterogeneous criterion values before aggregation, but they do not address the specific situation of multi-layer IoT systems in which one and the same quality criterion is partially realized at Edge, Fog, and Cloud simultaneously. In such cases, even normalized layer-level contributions may still inflate the integral score if they are added without first distributing the total responsibility for that criterion across layers. Optimization-oriented studies address search, refactoring, portfolio generation, and application placement in Fog/Edge infrastructures [9, 10], yet they also typically assume a fixed criterion scale and do not isolate the layer-induced inflation effect.

The closest structural analog to the proposed matrix $\boldsymbol{W}$ is the weight vector derived from AHP-based methods [6], in which pairwise comparisons among criteria yield a normalized priority distribution that sums to one. However, AHP operates across criteria, distributing importance among quality attributes for a fixed, single-layer alternative. It does not address the situation in which one criterion is realized simultaneously across multiple architectural layers: when the AHP weight for “security” equals 0.3, this scalar governs only how security competes against other criteria in the final ranking, not how the security contribution of the Edge layer relates to that of the Cloud. Consequently, applying AHP weights to a multi-layer IoT system without prior inter-layer reconciliation does not prevent the inflation of integral scores; it only rescales an already-inflated sum.

A similar limitation applies to QoS-aware aggregation approaches widely used in Edge-Fog-Cloud placement and service composition research [7, 11]. These approaches normalize individual quality of service (QoS) attributes before aggregating them via weighted sums or composite fitness functions, which correctly handle heterogeneous units across attributes. Yet normalization across attributes and normalization across layers are orthogonal operations: even fully normalized layer-level contributions still inflate the integral criterion score when summed without distributing the criterion's total “mass” among layers. The specific gap that $\boldsymbol{W}$ closes is therefore not addressed by criterion-weight derivation (AHP), trade-off identification (ATAM/CBAM [5]), or QoS normalization: none of these mechanisms distributes the unit responsibility for a single criterion across the architectural layers in which it is concurrently realized.

In this paper, double-counting is defined as the situation where a single quality criterion acquires an artificially inflated integral value because normalized Edge, Fog, and Cloud contributions are added without a prior distribution of the criterion’s total “mass” among layers. This effect distorts the comparison of alternatives, complicates cross-scenario comparability, and weakens the interpretability of integral scores.

The Layer-Responsibility Matrix *W*. Let L={Edge, Fog, Cloud} be the set of architectural layers of an IoT system, K the set of quality criteria, P the catalog of architectural patterns, and $S \subseteq P \times L$ an architectural portfolio consisting of pattern layer pairs. The scenario context is denoted C. For each criterion $k \in K$ and layer $l \in L$, let $c_l\left(k,S;C\right) \in \left[0,1\right]$ denote the normalized layer-level contribution, representing the total contribution of all patterns in S deployed at layer l toward criterion k as $\left(p,l\right) \in S$.

Without inter-layer reconciliation, the criterion can be scored by naive summation as

$$q_k^{\Sigma}(S,C) = \sum_{l \in L} c_l(k,S;C)\,. \qquad (1)$$

Formula (1) is the source of double-counting and, in what follows, is treated as the baseline uncorrected layer-wise additive aggregation. With three layers, q_k^{Σ} can reach 3 even though each c_j is normalized to $\left[0,1\right]$ and semantically represents the degree of fulfillment of a single criterion.

To eliminate this effect, we introduce the layer-responsibility matrix $W = [w_{kl}]$. The element $w_{kl} \in \left[0,1\right]$

indicates the share of the total responsibility for criterion k that falls on layer l in scenario C. The key requirement is row normalization as

$$\sum_{l \in L} w_{kl} = 1, \forall k \in K . \quad (2)$$

Formula (2) means that criterion k is treated as a single unit of responsibility, not multiplied when a multi-layer architecture is adopted, but distributed among Edge, Fog, and Cloud. Consequently, adding a new layer to the configuration does not automatically inflate the criterion's integral weight; it only changes the structure of its realization.

Based on $\boldsymbol{W}$, the corrected aggregated contribution of criterion k is defined as

$$q_k^{\boldsymbol{W}}(S,C) = \sum_{l \in L} w_{kl} c_l(k,S;C) . \quad (3)$$

Formula (3) is the central result of this paper. It ensures scenario-dependent yet mathematically consistent computation of criterion scores across layers. For practical use, the matrix $\boldsymbol{W}$ must be elicited from the scenario rather than assigned arbitrarily. The matrix $\boldsymbol{W}$ is instantiated for a given scenario C by a lightweight expert procedure. For each criterion k, the architect first identifies the layer at which the criterion is primarily realized, using three practical questions: where the critical control or response loop is closed, where the main computational or storage burden related to the criterion is located, and where the dominant risk of criterion degradation arises. The dominant layer is assigned the largest share of responsibility, while the remaining share is distributed between the supporting layers according to their secondary contribution. If the criterion is inherently distributed, the responsibility may be allocated more evenly. In all cases, the resulting row is normalized so that

$$\sum_{l \in L} w_{kl} = 1 .$$

As practical heuristics, low latency and real-time reaction typically shift responsibility toward Edge, scalability and global coordination toward Cloud, reliability under unstable connectivity toward Fog or Edge-Fog combinations, whereas security is often distributed across all three layers.

Properties of *W*. Property 1. Non-negativity and normalization. Since $w_{kl} \in [0,1]$ and $\sum_{l \in L} w_{kl} = 1$, the matrix $\boldsymbol{W}$ defines a valid responsibility distribution without negative or redundant coefficients.

Property 2. Score boundedness. If $c_l(k,S;C) \in [0,1]$, then $q_k^{\boldsymbol{W}}(S,C) \in [0,1]$. Indeed, $q_k^{\boldsymbol{W}}$ is a convex combination of values in [0, 1] and therefore cannot exceed the interval.

Property 3. Invariance with respect to the number of layers. If the semantic content of criterion k remains the same, increasing the number of layers does not automatically increase $q_k^{\boldsymbol{W}}$, because the total weight of the criterion remains fixed at 1.

Property 4. Scenario adaptivity. Matrix $\boldsymbol{W}$ is not a universal constant. Its values depend on scenario C, the system type, physical constraints, data-processing topology, and dominant risks. The same non-functional requirement may be distributed among layers in fundamentally different ways across different IoT domains.

Property 5. Interpretability. Each coefficient w_{kl} has a natural interpretation: it reflects the degree to which a specific layer contributes to achieving a criterion. This makes $\boldsymbol{W}$ suitable not only for computation but also for expert discussion and validation.

Numerical Example. To demonstrate that the proposed aggregation mechanism affects not only the magnitude of scores but also the ranking of architectural alternatives, consider a simplified IoT scenario: smart-building climate control with three quality criteria $\left(K = \{\text{low latency, reliability, energy efficiency}\}\right)$ and the standard three-layer decomposition $L = \{\text{Edge, Fog, Cloud}\}$. Two candidate portfolios are compared:

– Portfolio A ("cloud-centric"): comprises patterns with strong Cloud-side presence (e.g., API Gateway, Cloud Load Balancer, Remote Monitoring), yielding moderate-to-high contributions at every layer;

– Portfolio B ("edge-concentrated"): comprises patterns focused on the local control loop (e.g., Local Caching, Sense-Compute-Control, Watchdog Timer), yielding high Edge contributions but weak Cloud scores.

The layer-responsibility matrix $\boldsymbol{W}$ for this scenario (Table 1) is constructed according to the elicitation procedure described in Section 3. In a climate-control system, the sensor-actuator loop is closed at the Edge, so low latency and energy efficiency are Edge-dominated. Reliability depends primarily on local aggregation and failover mechanisms at the Fog gateway, with Cloud providing a secondary backup role.

Table 1 – Layer-responsibility matrix ***W*** for the smart-building climate-control example.

Criterion	Edge	Fog	Cloud
Low latency	0.60	0.25	0.15
Reliability	0.25	0.45	0.30
Energy efficiency	0.55	0.30	0.15

Table 2 presents the layer-level contributions c_j for both portfolios, together with the naive aggregation q_k^{Σ} (formula 1) and the corrected aggregation $q_k^{\boldsymbol{W}}$ (formula 3).

Under naive aggregation, Portfolio A dominates on every criterion (1.80 > 1.60, 1.90 > 1.70, 1.65 > 1.55) and achieves a higher total score (5.35 vs. 4.85). An architect relying on this ranking would select A. However, A's advantage is an artifact of inter-layer double-counting: its uniformly moderate contributions are amplified by summation across all three layers, particularly by the Cloud scores (0.70, 0.75, 0.65) that contribute to criteria for which Cloud bears only minor responsibility.

Table 2 – Comparison of two portfolios under naive and corrected aggregation

Criterion	c_E	c_F	c_C	q_k^{Σ}	q_k^{W}
Portfolio A					
Low latency	0.50	0.60	0.70	1.80	0.56
Reliability	0.55	0.60	0.75	1.90	0.63
Energy efficiency	0.45	0.55	0.65	1.65	0.51
Sum	–	–	–	5.35	1.70
Portfolio B					
Low latency	0.85	0.55	0.20	1.60	0.68
Reliability	0.50	0.80	0.40	1.70	0.61
Energy efficiency	0.80	0.60	0.15	1.55	0.64
Sum	–	–	–	4.85	1.93

After applying ***W***, the ranking reverses. Portfolio B outperforms A on two of the three criteria – low latency (0.677 vs. 0.555) and energy efficiency (0.643 vs. 0.510) – and yields a higher aggregate corrected score (1.925 vs. 1.698). Portfolio A retains an advantage only on reliability (0.633 vs. 0.605), where its stronger Cloud contribution (0.75 vs. 0.40) aligns with the non-negligible Cloud share of responsibility ($w_{\text{reliability}}, \text{Cloud} = 0.30$).

The reversal is not a numerical coincidence but a direct consequence of the domain logic encoded in ***W***. In a climate-control system, real-time response and energy management are determined primarily at the Edge, where the physical control loop operates. Portfolio B concentrates <u>its architectural strength precisely at this layer</u> ($c_E = 0.85$ for latency, 0.80 for energy), which is recognized by ***W*** through dominant Edge weights (0.60 and 0.55, respectively). Portfolio A distributes effort more evenly but fails to prioritize the layer that matters most – a weakness masked by naive summation but exposed by the corrected formula. Thus, the example demonstrates that ***W*** does not merely rescale scores but can also change the outcome of architectural decisions, favoring the portfolio whose structural emphasis aligns with the actual distribution of responsibilities in the target domain.

Case Study. The proposed method is examined under realistic application conditions through two case studies that represent different classes of IoT systems. The purpose of the analysis is not to repeat the computational search procedure itself, but to show how the layer-responsibility matrix ***W*** changes the interpretation of the obtained architectural result. For each scenario, a previously derived compromise portfolio from multi-objective synthesis is taken as the architectural outcome, after which the scenario-specific matrix ***W*** is constructed and the corrected aggregation is compared with the baseline layer-wise summation. This makes it possible to evaluate whether the proposed mechanism preserves a unified criterion scale and reflects the actual distribution of architectural responsibility across Edge, Fog and Cloud.

Infrastructure Monitoring and Control. The experimental outcome considered in this scenario is the previously obtained compromise architectural portfolio for the infrastructure monitoring and control problem. This portfolio comprises API Gateway, OAuth2, TLS, Circuit Breaker, Health Checks, Message Broker, and Publish–Subscribe, and is treated here as the architectural result to be analyzed through the proposed ***W***-based aggregation mechanism. The computational search procedure that produced this portfolio is reported in the authors' earlier work, where a pool of 16 architectural patterns was evaluated against five quality criteria, yielding 19448 candidate portfolios, 2010 feasible ones, and 510 Pareto-optimal alternatives. In the present paper, these search results are not recomputed; instead, the focus is placed on the scenario-specific construction of ***W*** and on the comparison between corrected aggregation and the baseline layer-wise summation for the obtained portfolio.

The domain logic of this scenario determines the scenario-specific ***W*** matrix (Table 3). For the "low latency" criterion, the greatest responsibility lies with the Edge layer, because latency is formed before data reaches the Cloud. For "scalability", the dominant share shifts to Cloud, where elastic resource allocation and horizontal scaling are realized.

Table 3 – Layer-responsibility matrix ***W*** for infrastructure monitoring

Criterion	Edge	Fog	Cloud
Security	0.20	0.35	0.45
Reliability	0.25	0.40	0.35
Low latency	0.65	0.25	0.10
Scalability	0.10	0.30	0.60
Complexity	0.30	0.40	0.30

Table 4 presents the layer-level contributions, cl, for the compromise portfolio of Scenario 1, along with both aggregation schemes. The layer-level contributions were derived from the pattern contribution profiles used in the computational experiment: for each layer, the contributions of all patterns assigned to that layer were summed and saturated to [0, 1].

Without ***W***, all criteria exceed 1 (ranging from 1.55 to 2.10), rendering them mutually incomparable on a single scale. After applying ***W***, scores return to [0, 1] and acquire meaningful interpretation. For "low latency," the corrected value of 0.710 is high precisely because the largest layer-level contribution is observed at Edge, and ***W*** assigns this

Table 4 – Aggregated scores for the compromise portfolio of Scenario 1

Criterion	c_E	c_F	c_C	No ***W***	With ***W***	Interpretation
Security	0.50	0.70	0.80	2.00	0.705	Inflated without ***W*** across 3 layers
Reliability	0.60	0.80	0.70	2.10	0.715	Balanced Fog-Cloud responsibility
Low latency	0.85	0.55	0.20	1.60	0.710	Edge-dominated criterion
Scalability	0.20	0.45	0.90	1.55	0.695	Cloud-dominated criterion
Complexity	0.35	0.75	0.60	1.70	0.585	Center of gravity at Fog

layer the dominant responsibility share (0.65). For “scalability”, the value 0.695 is driven primarily by the strong Cloud contribution, which is also consistent with the scenario logic.

Bionic Prosthesis Software. The experimental outcome considered in this scenario is the previously obtained compromise architectural portfolio for bionic prosthesis software. This portfolio comprises Graceful Degradation, Health Checks, Local Caching, Sense–Compute–Control, TLS, Token Auth, and Watchdog Timer, and is treated here as the architectural result to be analyzed through the proposed ***W***-based aggregation mechanism. The computational search procedure that produced this portfolio is reported in the authors’ earlier work, where a pool of 16 architectural patterns was evaluated against five quality criteria, yielding 23816 candidate portfolios, 3690 feasible ones, and 144 Pareto-optimal alternatives. In the present paper, these search results are not recomputed; instead, the focus is placed on the scenario-specific construction of ***W*** and on the comparison between corrected aggregation and the baseline layer-wise summation for the obtained portfolio.

The domain logic of a prosthesis shifts the center of responsibility sharply toward Edge (Table 5). Patient safety and energy efficiency both have over 50 % of their weight at Edge, reflecting the fact that bodily interaction and local control cannot be delegated to remote layers without unacceptable risk or latency. The matrices in Tables 3 and 5 were constructed according to the elicitation procedure described in Section 3.

Table 5 – Layer-responsibility matrix ***W*** for bionic prosthesis software

Criterion	Edge	Fog	Cloud
Patient safety	0.55	0.30	0.15
Latency	0.70	0.20	0.10
Reliability	0.40	0.35	0.25
Energy efficiency	0.75	0.20	0.05
Data security	0.35	0.25	0.40

Table 6 presents the layer-level contributions, cl, for the compromise portfolio of Scenario 2, along with both aggregation schemes. The layer-level contributions were derived from the pattern contribution profiles used in the computational experiment: for each layer, the contributions of all patterns assigned to that layer were summed and saturated to [0, 1].

Comparing Tables 4 and 6 reveals that ***W*** is not merely a scale corrector but a scenario-dependent model of architectural responsibility. For the prosthesis, the criteria “patient safety”, “latency”, and “energy efficiency” receive high corrected scores precisely because their strongest layer-level contributions are observed at Edge, and ***W*** assigns Edge the dominant share. For “data security”, the center of responsibility shifts to Cloud, reflecting centralized logging, access policies, and remote-update governance. The two ***W*** matrices are structurally different, yet the aggregation formula remains the same, confirming the domain-independence of formalism and the domain-specificity of its parametrization.

Discussion. The results lead to several generalizations. First, matrix ***W*** resolves a structural, not cosmetic, problem of additive models in multi-layer systems by correcting the baseline aggregation that otherwise conflates criterion fulfillment with the number of participating layers. Without it, the model conflates two distinct meanings: the degree of criterion fulfillment and the number of layers through which the criterion is realized. As a consequence, architectures with more distributed execution may automatically receive a higher integral score even if the actual quality of the solution is not superior.

Second, ***W*** preserves the simplicity of the additive model. Unlike non-linear or quadratic interaction models, it does not require a transition to a different class of computational methods. Aggregation remains linear, so the approach can be directly integrated into ranking schemes, multi-criteria portfolio synthesis models, and Pareto-set construction procedures [1, 12].

Third, ***W*** makes explicit the domain knowledge that often exists only implicitly in many methodologies. An architect typically understands that low latency for a prosthesis is primarily determined by the peripheral control loop, while the scalability of an analytics platform depends mainly on the Cloud. However, without a formal projection of this understanding into the model, it does not influence computation in a controlled and reproducible manner.

Fourth, the proposed approach is compatible with the broader formal apparatus developed in the authors’ earlier work [1]. In that framework, ***W*** combines with the representation of architecture as a portfolio $S \subseteq P \times L$, with contribution functions $c(k,p,l;C)$, with the coverage relation $R(C)$, and with multi-criteria synthesis and compromise-selection procedures. This means that ***W*** is not an isolated local idea but an organic part of a unified decision-support model.

Finally, it is important to delineate the scope of applicability. Matrix ***W*** eliminates specifically inter-layer double-counting. If within a single layer several patterns duplicate the same mechanism or exhibit strong synergistic/conflicting effects, ***W*** alone is insufficient. In such cases, compatibility constraints, pattern grouping, or a separate interaction-modeling layer are needed. Thus, ***W*** does not replace a full pattern-interaction model but substantially improves the baseline additive model, from which the problem originates.

The case studies show that ***W*** should be interpreted not merely as a normalization device, but as a scenario-dependent model of architectural responsibility. For both infrastructure monitoring and bionic prosthesis software, the same aggregation rule remained valid, whereas the responsibility matrices differed according to domain logic. This indicates that the proposed formalism is domain-independent at the level of computation, yet domain-sensitive at the level of parametrization.

Table 6 – Aggregated scores for the compromise portfolio of Scenario 2

Criterion	c_E	c_F	c_C	No ***W***	With ***W***	Interpretation
Patient safety	0.90	0.70	0.30	1.90	0.750	Critical local responsibility
Latency	0.88	0.50	0.25	1.63	0.741	Latency set at Edge
Reliability	0.72	0.78	0.60	2.10	0.711	Balanced across three layers
Energy efficiency	0.86	0.45	0.20	1.51	0.745	Local control loop dominates
Data security	0.40	0.55	0.80	1.75	0.598	Significant Cloud role

Consequently, ***W*** should be regarded not as an auxiliary correction device but as a first-class element of the architectural decision model, standing alongside the portfolio representation, the contribution functions, and the Pareto-synthesis procedure.

Limitations and Future Work. The proposed approach still has several limitations. First, although Section 3 provides a minimal elicitation procedure for constructing the matrix ***W***, its parametrization remains partly dependent on expert judgment and scenario interpretation. Different architects may assign slightly different responsibility shares even when considering the same system context. Future work should therefore investigate automated or semi-automated derivation of ***W*** from empirical data, such as telemetry, latency logs, energy-consumption profiles, or load-test results.

Second, ***W*** does not model interactions within-layer patterns. If two security tactics duplicate each other or, conversely, amplify each other's effect, this is not reflected directly in formula (3). For such cases, the model needs to be extended, or separate interaction coefficients introduced.

Third, the paper uses scenario illustrations for two domains but does not attempt to build a universal catalog of ***W*** matrices for all classes of IoT systems. Such a catalog could become a separate research direction and a practical artifact for architects [13].

Fourth, a potentially fruitful direction is to combine ***W*** with a posteriori selection procedure, in particular, with Pareto-alternative ranking via TOPSIS-like approaches or scenario-based risk analysis [9, 12]. In this case, ***W*** would serve as a mechanism for constructing the correct criterion prior to the final compromise selection stage.

Conclusions. The study has shown that additive quality models for multi-layer IoT systems contain a structural source of bias: when Edge, Fog, and Cloud contributions are summed directly, the resulting score depends not only on criterion fulfillment, but also on the number of layers in which the criterion is realized. As a result, naive aggregation loses a unified interpretation scale, making cross-criterion comparisons unreliable.

Results demonstrate that the layer-responsibility matrix ***W*** resolves this problem by separating two notions that are otherwise conflated in baseline additive models: the degree of criterion fulfillment and the structural distribution of its realization across layers. In this way, the corrected aggregation preserves boundedness and comparability of scores while retaining the simplicity and linearity of the additive model.

Declaration on the use of generative AI. During the preparation of this work, the authors used Claude (Anthropic) for grammar and spell checking, as well as for rephrasing and reformulating the text. After using this tool, the authors reviewed and edited the content as necessary and take full responsibility for the content of this publication.

References

1. Alfonso I., Garcés K., Castro H., Cabot J. Self-adaptive architectures in IoT systems: a systematic literature review. *Journal of Internet Services and Applications*. 2021, vol. 12, article 14. DOI: 10.1186/s13174-021-00145-8.
2. Ashouri M., Davidsson P., Spalazzese R. Quality attributes in edge computing for the Internet of Things: A systematic mapping study. *Internet of Things*. 2021, vol. 13, article 100346. DOI: 10.1016/j.iot.2020.100346.
3. Chumachenko D., Liubchenko V. A multi-objective optimization model for designing the software architecture of Internet of Things systems. *Herald of Khmelnytskyi National University. Technical Sciences*. 2026, vol. 363, no. 2, pp. 103–109. DOI: 10.31891/2307-5732-2026-363-13.
4. Jafari V., Rezvani M. H. Joint optimization of energy consumption and time delay in IoT-fog-cloud computing environments using NSGA-II metaheuristic algorithm. *Journal of Ambient Intelligence and Humanized Computing*. 2023, vol. 14, pp. 1675–1698. DOI: 10.1007/s12652-021-03388-2.
5. Bass L., Clements P., Kazman R. *Software Architecture in Practice*. 2nd ed. Addison-Wesley Longman, 2003.464 p.
6. Zhu L., Aurum A., Gorton I., Jeffery R. Tradeoff and Sensitivity Analysis in Software Architecture Evaluation Using Analytic Hierarchy Process. *Software Quality Journal*. 2005, vol. 13, pp. 357–375. DOI: 10.1007/s11219-005-4251-0.
7. Orellana C., Cereceda-Balic F., Solar M., Astudillo H. Enabling Design of Secure IoT Systems with Trade-Off-Aware Architectural Tactics. *Sensors*. 2024. Vol. 24, no. 22. Art. 7314. DOI: 10.3390/s24227314.
8. Vafaei N., Ribeiro R. A., Camarinha-Matos L. M. Selection of Normalization Technique for Weighted Average Multi-criteria Decision Making. *Camarinha-Matos L., Adu-Kankam K., Julashokri M., eds. Technological Innovation for Resilient Systems. DoCEIS 2018. IFIP Advances in Information and Communication*

Technology. *Vol. 521*. Springer, Cham, 2018, pp. 43–52. DOI: 10.1007/978-3-319-78574-5_4.
9. Di Pompeo D., Tucci M. Multi-objective Software Architecture Refactoring driven by Quality Attributes. *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*. 2023, pp. 175–178. DOI: 10.1109/ICSA-C57050.2023.00046.
10. Smolka S., Mann Z. Á. Evaluation of fog application placement algorithms: a survey. *Computing*. 2022, vol. 104, pp. 1397–1423. DOI: 10.1007/s00607-021-01031-8.
11. Tran M.-Q. et al. Energy Efficient Node Selection in Edge-Fog-Cloud Layered IoT Architecture. *Sensors*. 2023, vol. 23, no. 13, art. 6039. doi: 10.3390/s23136039
12. Lust T., Teghem J. The multiobjective multidimensional knapsack problem: A survey and a new approach. *International Transactions in Operational Research*. 2012, vol. 19, no. 4, pp. 495–520. DOI: 10.1111/j.1475-3995.2011.00840. x.
13. Oprea S. V., Bâra A. An Edge-Fog-Cloud computing architecture for IoT and smart metering data. *Peer-to-Peer Networking and Applications*. 2023, vol. 16, pp. 818–845. DOI: 10.1007/s12083-022-01436-y.

Received 11.03.2026
Accepted 07.04.2026
Published 20.05.2026

УДК 004.41 + 004.738.5 + 519.816

Д. К. ЧУМАЧЕНКО, аспірант кафедри інженерії програмного забезпечення, Національний університет «Одеська політехніка», пр. Шевченка, 1, м. Одеса, Україна, 65044; e-mail: chumachdk@stud.op.edu.ua; ORCID: https://orcid.org/0009-0000-1477-534X
В. В. ЛЮБЧЕНКО, доктор технічних наук, професор кафедри інженерії програмного забезпечення, Національний університет «Одеська політехніка», пр. Шевченка, 1, м. Одеса, Україна, 65044; e-mail: lvv@op.edu.ua; ORCID: https://orcid.org/0000-0002-4611-7832

АГРЕГУВАННЯ КРИТЕРІЇВ ЯКОСТІ В БАГАТОШАРОВИХ ПРОГРАМНИХ АРХІТЕКТУРАХ ІОТ З УРАХУВАННЯМ ВАГ ВІДПОВІДАЛЬНОСТІ

У статті розглянуто проблему подвійного врахування критеріїв якості в багатошарових програмних архітектурах Інтернету речей, де один і той самий нефункціональний критерій одночасно реалізується на рівнях Edge, Fog і Cloud. Показано, що традиційне адитивне підсумовування внесків окремих шарів породжує структурне викривлення оцінювання: інтегральне значення критерію починає залежати не лише від реального ступеня його досягнення, а й від кількості шарів, у яких цей критерій представлено. Це призводить до втрати єдиної інтерпретаційної шкали, ускладнює міжкритеріальне порівняння та може зміщувати архітектурний вибір на користь рішень, у яких ефект розподілено між більшою кількістю шарів. Для усунення цього недоліку запропоновано матрицю ваг відповідальності шарів, яка задає розподіл сумарної відповідальності за кожен критерій якості між архітектурними рівнями та використовується в процедурі зваженого агрегування. Виведено скориговану формулу агрегування, встановлено основні властивості запропонованого формалізму, зокрема невід’ємність, обмеженість, інваріантність щодо кількості шарів, сценарну адаптивність та інтерпретованість. Додатково подано мінімальну експертну процедуру формування матриці ваг на основі особливостей сценарію, контурів керування, локалізації обчислювального навантаження та домінантних ризиків. На числовому прикладі показано, що запропонований підхід не лише усуває інфляцію оцінок, а й здатний змінювати ранжування архітектурних альтернатив, тобто впливати на підсумковий вибір портфеля патернів. Практичну придатність підходу продемонстровано на двох сценаріях – системі моніторингу та керування інфраструктурою й програмному забезпеченні біонічного протеза. Отримані результати підтверджують, що зважене агрегування за відповідальністю шарів забезпечує більш коректне та змістовно узгоджене оцінювання багатошарових IoT-архітектур і може розглядатися як окремий компонент формального апарату підтримки архітектурних рішень.

Ключові слова: програмна архітектура IoT; багатошарові архітектури; критерії якості; адитивне агрегування; подвійне врахування; матриця ваг відповідальності; Edge-Fog-Cloud; підтримка архітектурних рішень.

Повні імена авторів / Author's full names

Автор 1 / Author 1: Чумаченко Данило Кирилович / Chumachenko Danylo Kyrylovych
Автор 2 / Author 2: Любченко Віра Вікторівна / Liubchenko Vira Viktorivna